

Complexity Results for Three-Dimensional Orthogonal Graph Drawing*

(Extended Abstract)

Maurizio Patrignani

Università di Roma Tre, Italy
patrigna@dia.uniroma3.it

Abstract. We introduce the 3SAT reduction framework which can be used to prove the NP-hardness of finding three-dimensional orthogonal drawings with specific constraints. We use it to show that finding a drawing of a graph whose edges have a fixed shape is NP-hard. Also, it is NP-hard finding a drawing of a graph with nodes at prescribed positions when a maximum of two bends per edge is allowed. We comment on the impact of these results on the two open problems of determining whether a graph always admits a 3D orthogonal drawing with at most two bends per edge and of characterizing orthogonal shapes admitting a drawing without intersections.

1 Introduction

Three-dimensional orthogonal graph drawing has attracted a constant research interest throughout the last decade [1, 4, 10, 11, 12, 13, 18, 20]. Nevertheless, some basic questions still lack an answer. It is open, for example, whether a graph of maximum degree six always admits a drawing with at most two bends per edge ([11, 20], and [5], problem #46). In [19] it is shown that such drawings may imply many edges sharing the same axis-perpendicular plane.

Also, a characterization of the orthogonal shapes admitting an orthogonal drawing without intersections (called *simple orthogonal shapes*) is still missing in the general case ([9] and [3], problem 20). Such a characterization would allow the separation of the task of defining the shape of the drawing from the task of computing its coordinates, extending to three-dimensions the well studied and widely adopted two-dimensional approach known as topology-shape-metrics [16, 6].

More formally, we would like to find the solution to the so-called *Simplicity Testing Problem*: Let G be a graph whose edges are directed and labeled with

* Work partially supported by European Commission – Fet Open project COSIN – COevolution and Self-organisation In dynamical Networks – IST-2001-33555, by European Commission – Fet Open project DELIS – Dynamically Evolving Large Scale Information Systems – Contract no 001907, by MIUR under Project ALGO-NEXT (Algorithms for the Next Generation Internet and Web: Methodologies, Design, and Experiments), and by “The Multichannel Adaptive Information Systems (MAIS) Project”, MIUR Fondo per gli Investimenti della Ricerca di Base.

a sequence of labels in the set $\{x+, x-, y+, y-, z+, z-\}$. Does a 3D orthogonal drawing of G exist such that each edge has a shape “consistent” with its labeling and no two edges intersect?

While a solution to the 2D counterpart of the Simplicity Testing Problem can be found in the works by Vijaian and Widgerson and by Tamassia [16, 17], only very preliminary results toward the recognition of simple orthogonal 3D shapes are provided in [7, 8] where paths (with further additional constraints) and cycles are considered, respectively. In [9] it is shown that the known characterization for cycles does not immediately extend to even seemingly simple graphs such as theta graphs (simple graphs consisting of three cycles).

In this paper we consider three-dimensional orthogonal drawing of a maximum degree six graph from the computational complexity perspective. We introduce the 3SAT reduction framework which can be used to show that it is NP-hard to decide if an orthogonal 3D drawing of a graph satisfying some constraints exists (Section 3). By using such a framework we show that the *simplicity testing* problem is NP-complete (Section 4) and that the opposite problem of finding a drawing of a graph with nodes at prescribed positions is also NP-complete when a maximum of two bends per edge is allowed (Section 5), while it is polynomial in the general case.

We comment on the impact of these results on the two open problems of determining whether a graph always admits a drawing with at most two bends per edge and of characterizing orthogonal shapes admitting an orthogonal drawing without intersections (Section 6).

2 Background

We assume familiarity with basic graph drawing, graph theory, and computational geometry terminology (see, e.g. [6, 2, 15]).

A (*3D orthogonal*) *drawing* of a graph is such that nodes are mapped to distinct points of the three dimensional space and edges are chains of axis-parallel segments. A *bend* is a point shared between two subsequent segments of the same edge. An *intersection* in a 3D orthogonal drawing is a pair of edges that overlap in at least one point that does not correspond to a common end-node. A *k-bend drawing* of a graph, where k is a non-negative integer, is a non-intersecting drawing such that each edge has at most k bends.

An x -plane (y -plane, z -plane, respectively) is a plane perpendicular to the x axis (y axis, z axis, respectively). Given a drawing Γ of a graph G and two nodes u and v , we write $u >_x v$ if the x coordinate of u is greater than the x coordinate of v in Γ . Also, we write $u >_{x>y} v$ if $u >_x v$ and $u >_y v$.

A *direction label* is a label in the set $\{x+, x-, y+, y-, z+, z-\}$. Let G be a graph and Γ be a drawing of G . Let e be an undirected edge of G whose end-nodes are u and v . Select one of the two possible orientations (u, v) and (v, u) of e and call p_1, p_2, \dots, p_m the end points of the orthogonal segments corresponding to edge e in Γ in the order in which they are encountered while moving along e from u to v . The *shape of e in Γ* is the sequence of the direction labels corresponding

to the directions of vectors $\overrightarrow{p_i, p_{i+1}}$, $i = 1, \dots, m - 1$. For example, consider an edge (u, v) drawn with a single bend b and such that $u <_x b <_y v$. The shape of e consists of the orientation from u to v and the sequence of labels $x+$, $y+$. We also write $u \xrightarrow{x+} \xrightarrow{y+} v$.

When producing a drawing of a graph one can ask if the positions of the vertices and the shapes of the edges can be computed separately. When computing positions first, one has to solve the following problem.

Problem: **ROUTING**

Instance: A graph $G(V, E)$ and a mapping between nodes and distinct points of the three-dimensional space.

Question: Does a non-intersecting 3D orthogonal drawing of G exist such that the nodes have the specified coordinates?

We call 2-BEND ROUTING the ROUTING problem when restricted to 2-bend drawings.

Conversely, it can be asked what is the complexity of deciding if a graph admits a drawing such that its edges have a specified shape. We call *shape graph* a graph where a shape (an orientation and a sequence of direction labels) is specified for each one of its edges. A shape graph γ is *simple* if it admits a non-intersecting drawing Γ such that each edge has the specified shape. Formally, the SIMPLICITY TESTING problem is as follows.

Problem: **SIMPLICITY TESTING**

Instance: A shape graph γ , that is, a graph $G(V, E)$ and a shape for each edge $e \in E$, consisting of an orientation of e and a sequence of labels in the set $\{x+, x-, y+, y-, z+, z-\}$.

Question: Does a non-intersecting drawing of G exist such that each edge has the specified shape?

3 The 3SAT Reduction Framework

The 3SAT reduction framework introduced in this section can be used to show that it is NP-hard finding a 3D drawing of a graph within the orthogonal standard that satisfies some constraints. By using this framework it is shown, in Sections 4 and 5, respectively, the NP-hardness of SIMPLICITY TESTING and of 2-BEND ROUTING. Throughout this section, the target problem is assumed to be as follows:

Problem: **Target problem**

Instance: A graph $G(V, E)$ and a set S of constraints expressed with respect to its nodes and edges.

Question: Does a non-intersecting 3D drawing of G exist such that the constraints in S are satisfied?

The 3SAT problem is as follows:

Problem: 3-Satisfiability (3SAT)

Instance: A set of clauses $\{c_1, c_2, \dots, c_m\}$, each containing three literals from a set of boolean variables $\{v_1, v_2, \dots, v_n\}$.

Question: Can truth values be assigned to the variables so that each clause contains at least one true literal?

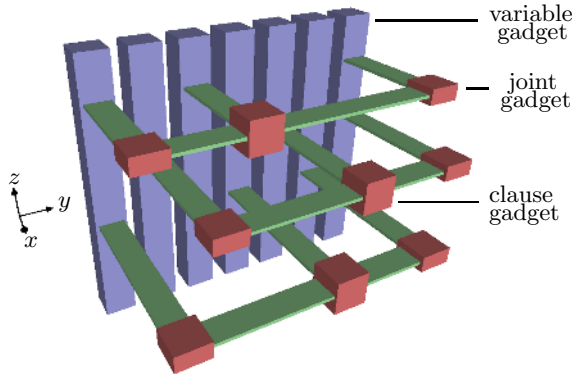


Fig. 1. A representation of the basic blocks of an instance of the target problem built as specified by the 3SAT reduction framework

Given a 3SAT instance ϕ , the 3SAT reduction framework specifies how to build an instance $I_\phi = (G_\phi(V_\phi, E_\phi), S_\phi)$ of the target problem such that ϕ admits a solution if and only if I_ϕ does. $G_\phi(V_\phi, E_\phi)$ is composed of three different types of gadgets connected together. The bounding boxes of the gadgets are depicted in Fig. 1, while the interior components are not shown and depend on the specific target problem.

For each boolean variable v_i of ϕ , instance I_ϕ has a *variable gadget* V_i . Fig. 1 shows the variable gadgets as tall vertical blocks placed in a row along the y axis in such a way that, if $i < j$, variable gadget V_i has lower y coordinates than variable gadget V_j .

For each clause $c_i = l_h \vee l_j \vee l_k$ of ϕ instance I_ϕ has one *clause gadget* C_i . Clause gadgets are represented in Fig. 1 as small cubes. Denoted with v_h, v_j , and v_k the variables of literals l_h, l_j , and l_k , respectively, and assumed that $h < j < k$, clause gadget C_i is placed directly in front of the variable gadget V_j .

For each clause $c_i = l_h \vee l_j \vee l_k$ of ϕ , I_ϕ has two *joint gadgets* $J_{i,h}$ and $J_{i,k}$, depicted in Fig. 1 as flat blocks placed in front of the variable gadgets V_h and V_k , respectively.

In order to use the 3SAT reduction framework for the NP-hardness proof of a specific target problem a complete specification must be provided, where a *specification* for the 3SAT reduction framework is defined as follows.

- Construction rules describing how, starting from an instance ϕ of the 3SAT problem, variable gadgets, joint gadgets, and clause gadgets are built and

connected together and an instance $I_\phi = (G_\phi(V_\phi, E_\phi), S_\phi)$ of the target problem is obtained.

- For each variable gadget V_i a bipartition of the non-intersecting drawings of $G_\phi(V_\phi, E_\phi)$ satisfying constraints S_ϕ into two sets, denoted T_{V_i} and F_{V_i} .
- For each joint gadget $J_{i,k}$ a bipartition of the non-intersecting drawings of $G_\phi(V_\phi, E_\phi)$ satisfying constraints S_ϕ into two sets, denoted $T_{J_{i,k}}$ and $F_{J_{i,k}}$.

A specification is said to be *compliant* if, for any 3SAT instance ϕ , the following four statements hold.

Statement 1. *Instance $I_\phi = (G_\phi(V_\phi, E_\phi), S_\phi)$ of the target problem corresponding to instance ϕ of 3SAT can be constructed in polynomial time.*

Statement 2. *If a non-intersecting drawing of $G_\phi(V_\phi, E_\phi)$ satisfying S_ϕ exists, it belongs to $T_{J_{i,h}}$ ($T_{J_{i,k}}$) if and only if it belongs to T_{V_h} (T_{V_k}).*

Statement 3. *For each clause $c_i = l_h \vee l_j \vee l_k$, where l_h (l_j , l_k , respectively) is the positive or the negative literal of variable v_h (v_j , v_k , respectively), and for each non-intersecting drawing Γ of $G_\phi(V_\phi, E_\phi)$ satisfying S_ϕ at least one among the following conditions holds:*

1. $\Gamma \in T_{J_{i,h}}$ ($\Gamma \in F_{J_{i,h}}$) and l_h is the positive (negative) literal of v_h .
2. $\Gamma \in T_{V_j}$ ($\Gamma \in F_{V_j}$) and l_j is the positive (negative) literal of v_j .
3. $\Gamma \in T_{J_{i,k}}$ ($\Gamma \in F_{J_{i,k}}$) and l_k is the positive (negative) literal of v_k .

Statement 4. *Consider a truth assignment to the variables v_1, \dots, v_n satisfying ϕ . The set $\bigcap_{i=1}^n A_i$, where $A_i = T_{V_i}$ if v_i is true and $A_i = F_{V_i}$ if v_i is false, is non-empty.*

Theorem 1. *Given a target problem, whose instance is a graph $G(V, E)$ and a set S of constraints expressed with respect to its nodes and edges, if it admits a compliant specification for the 3SAT reduction framework, then finding a non-intersecting 3D orthogonal drawing of G satisfying the constraints in S is NP-hard.*

Proof sketch. Consider a non-intersecting drawing Γ of $G_\phi(V_\phi, E_\phi)$ satisfying S_ϕ . It is easy to find an assignment of truth values to the boolean variables that satisfies ϕ , by taking $v_i = \text{true}$ if $\Gamma \in T_{V_i}$ and $v_i = \text{false}$ if $\Gamma \in F_{V_i}$. In fact, because of Statements 2 and 3 we have that each clause $c_i = l_h \vee l_j \vee l_k$ has at least one true literal and thus ϕ is satisfied. Conversely, consider an assignment of truth values to the boolean variables that satisfies ϕ . Statement 4 guarantees the existence of a drawing of $G_\phi(V_\phi, E_\phi)$ satisfying S_ϕ . The proof is completed by showing that instance $I_\phi = (G_\phi(V_\phi, E_\phi), S_\phi)$ can be obtained in polynomial time, which is guaranteed by Statement 1. \square

4 Fixing the Shape and Searching for Coordinates

In this section we consider the SIMPLICITY TESTING problem, that is the problem of finding a non-intersecting drawing for a graph whose orthogonal shape is fixed.

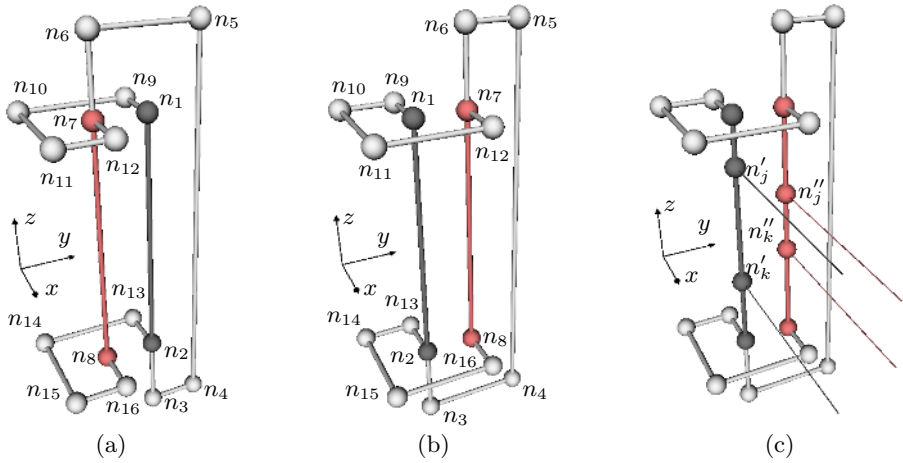


Fig. 2. A drawing of the variable gadget V_i belonging to T_{V_i} (a) and a drawing belonging to F_{V_i} (b). Nodes n'_j , n''_j , n'_k , and n''_k are inserted in order to transmit the geometric constraints to the clause gadgets of clauses C_j and C_k , respectively (c).

It is easy to prove that the problem is in NP (see [14]). In order to prove that SIMPLICITY TESTING is NP-hard we use the framework introduced in Section 3.

The heart of the variable gadget V_i , depicted in Fig. 2, is the path $n_1 \xrightarrow{z^-} n_2 \xrightarrow{z^-} n_3 \xrightarrow{y^+} n_4 \xrightarrow{z^+} n_5 \xrightarrow{y^-} n_6 \xrightarrow{z^-} n_7 \xrightarrow{z^-} n_8$, whose nodes lie on the same x -plane. Further, the path $n_1 \xrightarrow{x^+} n_9 \xrightarrow{y^-} n_{10} \xrightarrow{x^+} n_{11} \xrightarrow{y^+} n_{12} \xrightarrow{x^-} n_7$ constrains nodes n_1 and n_7 to share the same z -plane. Analogously, path $n_2 \xrightarrow{x^-} n_{13} \xrightarrow{y^-} n_{14} \xrightarrow{x^+} n_{15} \xrightarrow{y^+} n_{16} \xrightarrow{x^-} n_8$ constrains nodes n_2 and n_8 to share the same z -plane. We define T_{V_i} as the set of non-intersecting drawings of $G_\phi(V_\phi, E_\phi)$ satisfying the directions constraints and such that $n_1 >_y n_7$ (as in Fig. 2.a). Analogously, we define F_{V_i} as the set of non-intersecting drawings of $G_\phi(V_\phi, E_\phi)$ satisfying the direction constraints and such that $n_1 <_y n_7$ (as in Fig. 2.b). Observe that T_{V_i} and F_{V_i} form a bipartition of the non-intersecting drawings of $G_\phi(V_\phi, E_\phi)$ satisfying the direction constraints.

For each clause c_j of the 3SAT formula in which the variable participates we insert a node n'_j between nodes n_1 and n_2 and a node n''_j between nodes n_7 and n_8 . In any drawing Γ of $G_\phi(V_\phi, E_\phi)$ satisfying the direction constraints, nodes n'_j and n''_j have the same relative position with respect to the y axis as n_1 and n_7 , i.e., $n'_j >_y n''_j$ if $\Gamma \in T_{V_i}$ and $n'_j <_y n''_j$ if $\Gamma \in F_{V_i}$. Suitable edges attached to the nodes n'_j and n''_j along the protruding lines shown in Fig. 2.c transmit the above constraints from V_i to the clause gadget C_j (possibly via joint gadget $J_{j,i}$). Note that nodes n'_j and n''_j do not need to lie on the same z -plane.

Given a clause $c_i = l_h \vee l_j \vee l_k$, the joint gadget $J_{i,k}$ is the reflected image with respect to the y axis of the joint gadget $J_{i,h}$. Thus, in the following we will only describe the joint gadget $J_{i,h}$, which is depicted in Fig. 3 and composed of two cycles $\alpha = n_1 \xrightarrow{y^+} n_2 \xrightarrow{x^-} n_3 \xrightarrow{y^-} n_4 \xrightarrow{x^+} n_5 \xrightarrow{y^-} n_6 \xrightarrow{x^+} n_7 \xrightarrow{y^+} n_8 \xrightarrow{x^-} n_1$, and $\alpha' = n'_1 \xrightarrow{y^-} n'_2 \xrightarrow{x^+} n'_3 \xrightarrow{y^+} n'_4 \xrightarrow{x^-} n'_5 \xrightarrow{y^+} n'_6 \xrightarrow{x^-} n'_7 \xrightarrow{y^-} n'_8 \xrightarrow{x^+} n'_1$. Nodes n_1 and n'_1

are connected by a path $n_1 \xrightarrow{z^-} n''_1 \xrightarrow{z^-} n'_1$ while nodes n_5 and n'_5 are connected by the path $n_5 \xrightarrow{z^-} n''_5 \xrightarrow{z^-} n'_5$.

We define $T_{J_{i,h}}$ as the set of non-intersecting drawings of $G_\phi(V_\phi, E_\phi)$ satisfying the directions constraints and such that $n''_5 <_x n''_1$ (as in Fig. 3.a). Analogously, we define $F_{J_{i,h}}$ as the set of non-intersecting drawings of $G_\phi(V_\phi, E_\phi)$ satisfying the direction constraints and such that $n''_5 >_x n''_1$ (as in Fig. 3.b). Observe that, since in any non-intersecting drawing of $G_\phi(V_\phi, E_\phi)$ n''_1 and n''_5 have distinct x and y coordinates, $T_{J_{i,h}}$ and $F_{J_{i,h}}$ form a bipartition of the non-intersecting drawings of $G_\phi(V_\phi, E_\phi)$ satisfying the direction constraints.

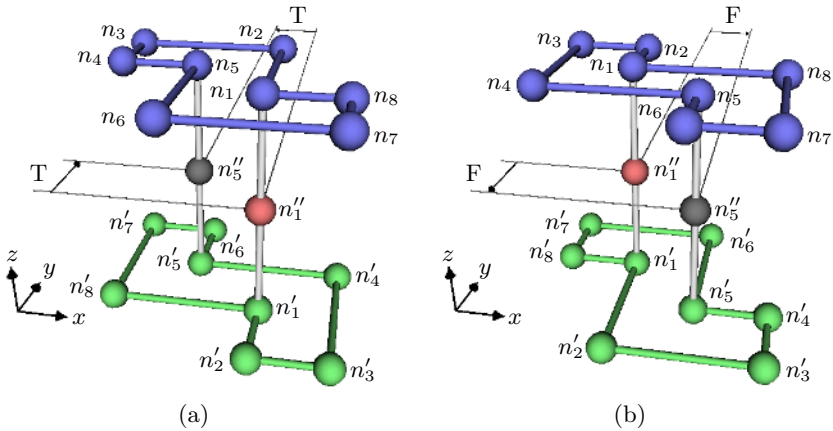


Fig. 3. A drawing of the joint gadget $J_{i,h}$ belonging to $T_{J_{i,h}}$ (a) and a drawing belonging to $F_{J_{i,h}}$ (b)

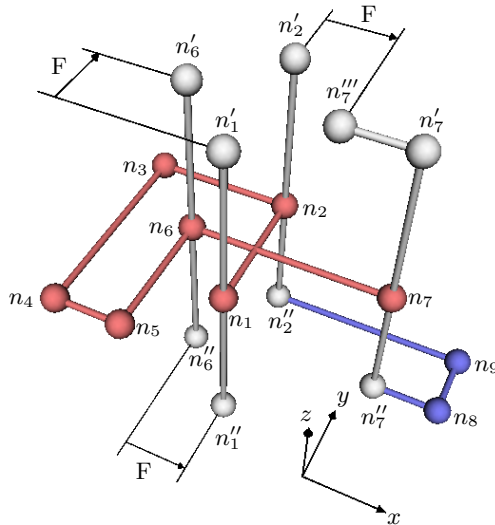


Fig. 4. Clause gadget

The clause gadget is depicted in Fig. 4. Its main component is the path $\alpha = n_1 \xrightarrow{y^+} n_2 \xrightarrow{x^-} n_3 \xrightarrow{y^-} n_4 \xrightarrow{x^+} n_5 \xrightarrow{y^+} n_6 \xrightarrow{x^+} n_7$, whose nodes lie on the same z -plane. Attached to α are the paths $n'_1 \xrightarrow{z^-} n_1 \xrightarrow{z^-} n''_1$, $n'_2 \xrightarrow{z^-} n_2 \xrightarrow{z^-} n''_2$, $n'_6 \xrightarrow{z^-} n_6 \xrightarrow{z^-} n''_6$, and $n'''_7 \xrightarrow{x^+} n'_7 \xrightarrow{z^-} n_7 \xrightarrow{z^-} n''_7 \xrightarrow{x^+} n_8 \xrightarrow{y^+} n_9 \xrightarrow{x^-} n''_2$.

Joint gadget $J_{i,h}$ is connected to both variable gadget V_h and clause gadget C_i . In particular, n''_1 of $J_{i,h}$ is connected to n''_i of V_h with the edge $n''_i \xrightarrow{x^+} n''_1$ and n''_5 of $J_{i,h}$ is connected to n'_i of V_h with the edge $n'_i \xrightarrow{x^+} n''_5$. Each clause $c_i = l_h \vee l_j \vee l_k$ is connected to joint gadget $J_{i,h}$, variable gadget V_j , and joint gadget $J_{i,k}$. If l_h is the positive (negative) literal of variable v_h , we attach nodes n''_1 and n''_5 of the joint gadget $J_{i,h}$ to nodes n''_6 and n''_1 (n'_1 and n'_6), respectively. If l_j is the positive (negative) literal of variable v_j , we attach nodes n'_i and n''_i of the variable gadget V_j to n'_6 and n'_1 (n'_1 and n'_6), respectively. If l_k is the positive (negative) literal of variable v_k , we attach nodes n_1 and n_4 of the joint gadget $J_{i,k}$ to nodes n'_2 and n''_7 (n''_7 and n'_2), respectively.

It is now easy to prove that the above construction rules are a compliant specification for the 3SAT reduction framework. Hence, we have:

Theorem 2. SIMPLICITY TESTING is NP-complete.

5 Fixing the Coordinates and Searching for a Shape

In this section we tackle the reverse problem with respect to the one addressed in Section 4, that is, the problem of finding a routing for the edges when the position of the nodes is fixed. An algorithm to solve ROUTING in $O(|V| \log |V|)$ time, where $|V|$ is the number of vertices of the input graph, can be found in [14]. Conversely, we show that the same problem where only two bends per edge are allowed (2-BEND ROUTING) is NP-complete.

In order to show that 2-BEND ROUTING is NP-hard we take advantage of the 3SAT reduction framework introduced in Section 3. The basic gadget shown in Fig. 5 is used as a building block of several parts of the 2-BEND ROUTING instance and is composed of ten nodes. Node n_1 is connected to the three nodes n_2, n_3 and n_4 . Analogously, node n_5 is connected to the three nodes n_6, n_7 and n_8 . Nodes n_1 and n_5 are connected both with the single edge (n_1, n_5) and with the path of three edges $(n_1, n_{1,5}), (n_{1,5}, n_{5,1})$ and $(n_{5,1}, n_5)$.

As for nodes prescribed positions, they are placed in such a way that $n_1 <_x <_y <_z n_2 =_x =_y <_z n_3 =_x =_y <_z n_4, n_1 =_x >_y >_z n_{1,5} =_x >_y >_z n_{5,1} =_x >_y >_z n_5$, and $n_5 <_x >_y >_z n_6 =_x =_y >_z n_7 =_x =_y >_z n_8$.

Given a 2-bend drawing of the basic gadget, we call *true* the basic gadget when it is drawn with the bend of edge (n_1, n_5) placed in $p_{t,2}$ (see Fig. 5.a) and *false* the basic gadget when it is drawn with the bend of edge (n_1, n_5) placed in $p_{f,2}$ (see Fig. 5.b). Also, in what follows we use the graphic representation of the basic gadget shown in Fig. 5.c, where the nodes n_1, n_2, n_3, n_4 , and $n_{1,5}$ are replaced by their bounding box, and analogously for the nodes n_5, n_6, n_7, n_8 , and $n_{5,1}$. In this representation only edge (n_1, n_5) is shown, and it is assumed to have its bend in $p_{t,2}$.

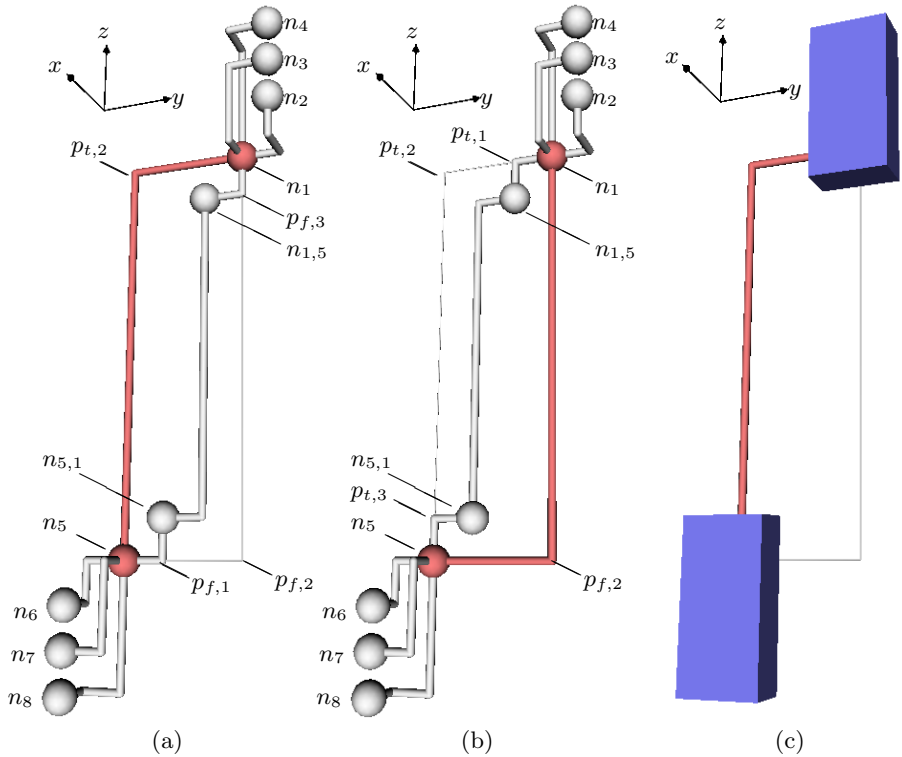


Fig. 5. A true drawing (a) and a false drawing (b) of the basic gadget. In (c) it is shown the schematic representation of the basic gadget that is used in the remaining part of the paper.

The variable gadget V_i is composed of a single basic gadget. Given a variable gadget V_i , we define as $T_{V_i}(F_{V_i})$ the set of non-intersecting 2-bend drawings of $G_\phi(V_\phi, E_\phi)$ such that the basic gadget is true (false).

The joint gadget $J_{i,h}$, which is depicted in Fig. 6 is built by interleaving four basic gadgets B_1, B_2, B_3 , and B_4 as follows. B_1 intersects the variable gadget (not shown in Fig. 6). B_2 is placed on an orthogonal plane as shown in Fig. 6.

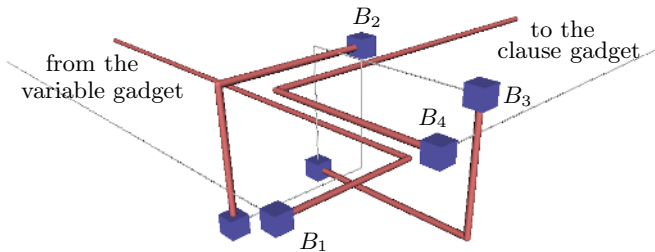


Fig. 6. Joint gadget $J_{i,h}$ is composed of four interleaved basic gadgets

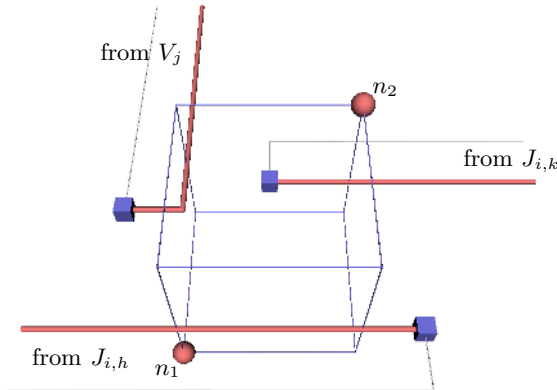


Fig. 7. The clause C_i for clause $c_i = l_h \vee l_j \vee l_k$ in the case in which l_h, l_j and l_k are negative literals: if the three variables v_h, v_h and v_k are true, there is no way of adding edge (n_1, n_2) with at most two bend

B_3 intersects only B_2 and is placed on a plane orthogonal to the first two (see Fig. 6). Finally, B_4 is placed on a plane parallel to the first one and intersects B_3 only as shown in Fig. 6. We define $T_{J_{i,h}} (F_{J_{i,h}})$ as the set of non-intersecting 2-bend drawings of G_ϕ satisfying S_ϕ such that B_4 is true (false).

The clause C_i for clause $c_i = l_h \vee l_j \vee l_k$ is shown in Fig. 7. It is composed of two nodes n_1 and n_2 placed at the opposite vertices of a cube. The two nodes are joined by edge (n_1, n_2) (not shown in Fig. 7). In any 2-bend drawing of the clause gadget edge (n_1, n_2) uses one of the four vertical edges of the cube. The basic gadget B_4 of joint gadgets $J_{i,h}$ and $J_{i,k}$ and the basic gadget coming from V_j suitably intersect the vertical edges of the cube such that only if one literal is true the clause gadget admits a non-intersecting drawing.

It is easy to show that 2-BEND ROUTING is in NP. Since the above described construction rules are a compliant specification for the 3SAT reduction framework, we have:

Theorem 3. 2-BEND ROUTING is NP-complete.

6 Discussion and Open Problems

This paper shows that SIMPLICITY TESTING is NP-complete, while the reverse problem, ROUTING, is feasible. This asymmetry may explain why most three-dimensional drawing algorithms in the literature determine edge shapes as a consequence of node relative positions and not vice versa.

With respect to the problem of characterizing simple orthogonal shapes, deciding whether a shape graph is simple is shown here to be NP-complete. Of course, the problem of characterizing simple orthogonal shapes remains open, although we now know that in the general case it implies a heavy computation.

As a consequence of the complexity of the SIMPLICITY TESTING problem in the general case, in any hypothetical 3D drawing process in which the definition of the shape of the drawing is followed by the actual computation of its coordinates, the first step should be very carefully conceived in order for the second step to be efficiently computable. In fact, focusing on peculiar classes of shape graphs seems to be an obliged strategy for practical applications. Are there non trivial families of shape graph for which the simplicity testing is feasible? In particular, is there a “universal” set of shape graphs such that any graph is represented and such that the simplicity testing is guaranteed to be polynomial and to have a positive answer?

With respect to the problem of determining if a graph of degree six always admits a 2-bend drawing, this paper shows the NP-completeness of two problems related with finding such drawings. Namely, it is NP-complete when node positions are fixed (Section 4) and it is NP-complete when edge shapes are fixed (Section 5). Some other 3D drawing problems involving the number of the bends are known to be NP-complete, as, for example, finding a 2-bend drawing when vertices are placed on the diagonal of a cube [21] (provided that the graph admits such a drawing). The number of NP-complete problems related with the computation of a 2-bend drawing raises the following question: What is the complexity of finding a 2-bend drawing of a graph? If finding such a drawing was also NP-hard, then any attempt to prove that such a drawing always exists should produce an algorithm for an intractable problem, which is hard to conceive without resorting to an enumerative approach (which, in turn, assumes the existence of a solution). However both the conception of such an algorithm and the description of a graph not admitting a 2-bend drawing appear to be elusive goals.

References

1. T. C. Biedl. Heuristics for 3D-orthogonal graph drawings. In *Proc. 4th Twente Workshop on Graphs and Combinatorial Optimization*, pages 41–44, 1995.
2. J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. Macmillan, London, 1976.
3. F. Brandenburg, D. Eppstein, M. T. Goodrich, S. Kobourov, G. Liotta, and P. Mutzel. Selected open problems in graph drawing. In G. Liotta, editor, *Graph Drawing (Proc. GD 2003)*, volume 2912 of *LNCS*, pages 515–539. Springer-Verlag, 2004.
4. M. Closson, S. Gartshore, J. Johansen, and S. K. Wismath. Fully dynamic 3-dimensional orthogonal graph drawing. *J. of Graph Algorithms and Applications*, 5(2):1–34, 2001.
5. E. D. Demaine, J. S. B. Mitchell, and J. O’Rourke, (eds.). The Open Problems Project. <http://cs.smith.edu/~orourke/TOPP/Welcome.html>.
6. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.
7. G. Di Battista, G. Liotta, A. Lubiw, and S. Whitesides. Orthogonal drawings of cycles in 3d space. In J. Marks, editor, *Graph Drawing (Proc. GD ’00)*, volume 1984 of *LNCS*. Springer-Verlag, 2001.

8. G. Di Battista, G. Liotta, A. Lubiw, and S. Whitesides. Embedding problems for paths with direction constrained edges. *Theor. Comp. Sci.*, 289:897–917, 2002.
9. E. Di Giacomo, G. Liotta, and M. Patrignani. A note on 3D orthogonal drawings with direction constrained edges. *Inform. Process. Lett.*, 90:97–101, 2004.
10. P. Eades, C. Stirk, and S. Whitesides. The techniques of Kolmogorov and Bardzin for three dimensional orthogonal graph drawings. *Inform. Process. Lett.*, 60:97–103, 1996.
11. P. Eades, A. Symvonis, and S. Whitesides. Three dimensional orthogonal graph drawing algorithms. *Discrete Applied Math.*, 103(1-3):55–87, 2000.
12. B. Y. S. Lynn, A. Symvonis, and D. R. Wood. Refinement of three-dimensional orthogonal graph drawings. In J. Marks, editor, *Graph Drawing (Proc. GD '00)*, volume 1984 of *LNCS*, pages 308–320. Springer-Verlag, 2001.
13. A. Papakostas and I. G. Tollis. Algorithms for incremental orthogonal graph drawing in three dimensions. *J. of Graph Algorithms and Applications*, 3(4):81–115, 1999.
14. M. Patrignani. Complexity results for three-dimensional orthogonal graph drawing. Tech. Report RT-DIA-94-2005, Dip. Inf. e Automazione, Univ. Roma Tre, 2005. <http://dipartimento.dia.uniroma3.it/ricerca/rapporti/rapporti.php>.
15. F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 3rd edition, Oct. 1990.
16. R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.
17. G. Vijayan and A. Wigderson. Rectilinear graphs and their embeddings. *SIAM J. Comput.*, 14:355–372, 1985.
18. D. R. Wood. On higher-dimensional orthogonal graph drawing. In J. Harland, editor, *Proc. Computing: the Australasian Theory Symposium (CATS '97)*, volume 19, pages 3–8. Australian Computer Science Commission, 1997.
19. D. R. Wood. Lower bounds for the number of bends in three-dimensional orthogonal graph drawings. *J. of Graph Algorithms and Applications*, 7:33–77, 2003.
20. D. R. Wood. Optimal three-dimensional orthogonal graph drawing in the general position model. *Theor. Comp. Sci.*, 299:151–178, 2003.
21. D. R. Wood. Minimising the number of bends and volume in 3-dimensional orthogonal graph drawings with a diagonal vertex layout. *Algorithmica*, 39:235–253, 2004.